3B-1

アクティビティ図で記述されたシステム運用操作からの可用性モデル合成方式

但野　紅美子†　向　剣文†　前野　義晴†

日本電気株式会社　サービスプラットフォーム研究所†

１．はじめに

企業の基幹システム等の高可用性が求められる情報システムにおける、情報システムの可用性の効率的な評価と改善の重要性が高まっている。情報システムの可用性に大きな影響を及ぼす要因の一つとして、情報処理システムを管理するための運用手順が挙げられる。適切なシステム運用手順の設計によって可用性を高めるためには、運用手順に含まれる多様な運用操作（システムの設定変更等）が可用性に与える影響を定量的に評価する必要がある。

２．関連研究

運用操作が可用性に与える影響を定量評価する手法として、モデルベースの可用性評価手法[1]が挙げられる。[1]は、管理対象システムに対する特定の運用操作に対応する可用性モデルを Stochastic Reward Nets(SRN) [2]を用いて構築し、構築した SRN モデルを解析することで可用性を推定する。しかしながら、このようにユーザが運用操作ごとに個別の可用性モデルを構築する評価手法は、下記の二つの問題点があった。第一に、実際の運用手順には多くの種類の運用操作が含まれるため、運用操作ごとに個別に可用性モデルの構築を繰り返すのは、煩雑で効率的ではない。第二に、モデル構築のためには専門知識が必要となるため、実際に運用手順の設計者自身が正しい可用性モデルを構築して評価に用いるのは困難である。

一方、ユーザが直接可用性モデルを構築するのではなく、システムの仕様が記述されたヒューマンフレンドリな UML 図[3]から可用性/信頼性モデルを自動生成する従来手法がある[4]。しかしながら、[4]で生成されるモデルは運用手順に含まれる多様な運用操作が可用性に与える影響を定量的に評価することはできない。本稿では、これらの問題を解決するために、ヒューマンフレンドリなアクティビティ図で記述された運用手順を SRN モデルに半自動的に変換する可用性モデル生成手法を提案する。

Synthesizing availability model from activity diagrams for system operations

†Kumiko Tadano, Jianwen Xiang, and Yoshiharu Maeno
Service Platforms Research Laboratories, NEC Corporation

３．提案手法

図１に、提案する可用性モデル生成手法の概念図を示す。本手法における可用性モデルは、運用手順モデルとシステム構成モデルの２つの SRN モデルから構成される。運用手順モデルは、評価対象となる運用手順の状態変化を表す。システム構成モデルは、操作対象システムコンポーネント(仮想/物理マシン等)の状態変化を表す。

本手法は、アクティビティ図変換機能とシステム構成モデル修正機能から構成される(図１中央)。アクティビティ図変換機能は、運用手順を記述したアクティビティ図を受け付け、運用手順モデルに変換する。システム構成モデル修正機能は、入力されたシステム構成モデルを修正し、アクティビティ図変換機能が生成した運用手順モデルの状態変化によって引き起こされるシステム構成モデルの状態変化を追加する。

本稿では、入力されたアクティビティ図からの運用手順モデルの生成に焦点を絞り、システム構成モデルについては稼働中/停止中の２状態を持つ SRN モデルの入力を仮定する。本稿では、システム構成モデルが稼働中の状態である確率を稼働率と定義する。より複雑な状態を持つシステム構成モデルの扱いは今後の課題とする。

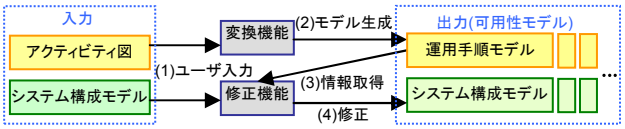

図１　可用性モデル生成手法の概念図

３．１　アクティビティ図変換機能

アクティビティ図変換機能の動作を、図２に示す。最初に、入力されたアクティビティ図を分割して部品化する。次に、部品ごとに変換規則に基づき SRN で表される運用手順モデルの部品に変換する。本手法では、一つのアクティビティと条件分岐の組を単位として、この単位ごとに一組の操作モデルと制御フローモデルに変換する。操作モデルは、一つの操作実行に伴う状態変化の詳細を記述し、実行結果を決定する（３．２節にて詳述）。制御フローモデルは、一つの操作の実行と実行結果に伴う制御の流れを表し、元のアクティビティ図の流れに従って接続される。トークンは現在の運用手順の状態

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.

を表すものとし、初期状態としてプレース P_{clock} と P_{wait} に各一個ずつ配置する。運用手順の定期的な実行を、一定時間 t_p ごとに発火するトランジション T_{clock} とトークンが $P_{startup}$ に入った際に T_{start} を発火可能にするガード関数 g_4 で表す。各操作が成功した場合トークンは次の操作へ進み、失敗した場合平均所要時間 t_{alert} で P_{start} に戻る。最後に、変換した運用手順モデルの部品を結合することで、運用手順モデルを生成する。

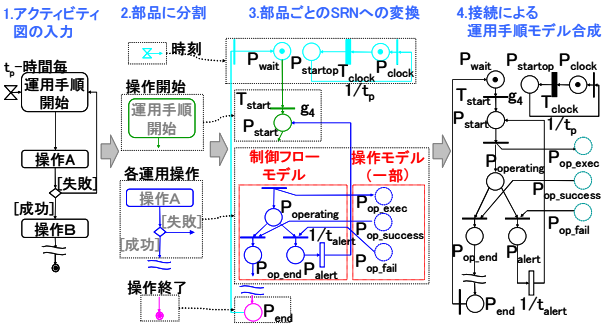

図２　アクティビティ図から SRN への変換

３．２　操作モデル

本手法の操作モデルの詳細について述べる。本手法は、運用手順に含まれる多様な運用操作を可用性に与える影響の観点から分類し、以下の抽象的な操作モデル(a)、(b)に集約することで、各操作に個別の操作モデルの設計を不要にする。(a)は操作対象の状態を変えない操作(リソースの使用状況や設定情報の表示等)、(b)は操作対象の状態を変える可能性がある操作(システムの設定変更等)に相当する。

運用操作について、以下の条件を想定する。(i)操作対象コンポーネントが停止中の場合、操作は必ず失敗する。(ii)操作は失敗する可能性がある。(iii)(b)の操作の失敗は、操作対象コンポーネントの障害を引き起こす可能性がある。

図３に、操作モデル(a)、(b)と各々の操作対象のシステム構成モデルを示す。ここでは、トークンがプレース P_{up} にある場合を稼働中とみなす。条件(i)より、もし対象システムコンポーネントが停止中の場合、ガード関数 g_3 がトランジション T_{down} を発火可能にする。稼働中の場合は、g_2 が T_{up} を発火可能にする。条件(ii)より、操作の成功率を c_{op} とおく。さらに(b)では、条件(iii)より、操作がもたらす操作対象への影響として、システム構成モデル修正機能が T_{fail} と g_1 を操作対象となるシステム構成モデルに追加する。g_1 は、操作失敗時に確率 $c_{sysfail}$ でトークンが P_{sv_fail} に入った際、T_{fail} を発火可能にする。

ユーザは、可用性モデルに必要な情報として、(1)各操作の種類、即ち、(a)または(b)、(2)各操作と操作対象システム構成モデルの対応関係、(3)パラメータの値(t_p, c_{op}, t_{alert}, $c_{sysfail}$, 平均所要時間 t_{op})を入力する。ユーザは、操作の特性に応じてこれらの値を変更するのみで、様々な操作に対する可用性モデルを生成可能となる。ユーザは、生成した可用性モデルを SPNP[5]等のツールを用いてトークンが P_{up} にある確率を解析することで、可用性を評価できる。

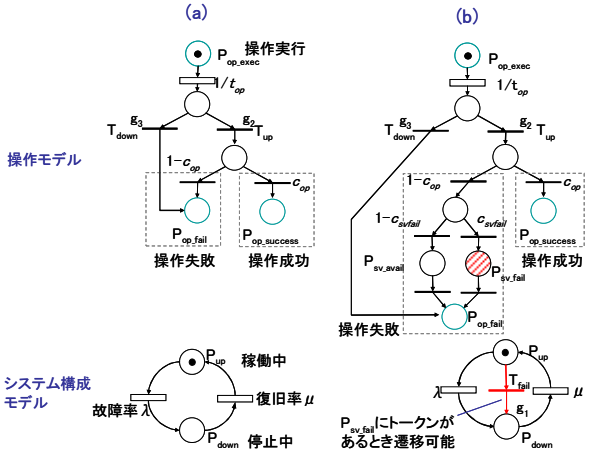

図３　操作モデル(上)と操作対象のシステム構成モデル（下）

４．まとめと今後の課題

本稿では、アクティビティ図で記述された運用手順を SRN モデルに半自動的に変換する可用性モデル生成手法について述べた。本手法では、運用手順に含まれる多様な運用操作が可用性に与える影響を２種類の操作モデルに集約することで、各操作に個別のモデル設計を不要にする。

今後の課題は、(1)実際の運用手順書に基づくアクティビティ図を用いた本手法の評価、(2)本手法が適用可能な操作の種類の拡大、(3)障害対応と定常運用に必要なコストの両方を含めた運用コストの評価、(4)本手法へのシステム構成モデルの自動生成手法の導入である。

参考文献

[1] V. Castelli et al., Proactive management of software aging, IBM Journal of Research and Development, Vol. 45, pp. 311-332, 2001.

[2] K. S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, John Wiley, New York, 2001.

[3] OMG Unified Modeling Language (OMG UML), Superstructure Version 2.3, http://www.omg.org/spec/UML/2.3/

[4] A. Bondavalli et al., Automated Dependability Analysis of UML Designs. In Proc. 2nd Int. Symp. on Objectoriented Real-time distributed Computing (ISORC'99), 1999.

[5] C. Hirel et al, "SPNP: Stochastic Petri Nets. Version 6.0," Proc. of TOOLS 2000.

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.